1991-09

Analysis of training-related issues in the transition to Ada in the DON.

Shkapsky, Jean Marie

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYSIS OF TRAINING-RELATED ISSUES
IN THE TRANSITION TO ADA IN THE DON

by

Jean Marie Shkapsky

September 1991

Thesis Advisor: LCDR Roger Stemp
Thesis Co-Advisor: Prof. Tung X. Bui

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable): AS	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
ANALYSIS OF TRAINING-RELATED ISSUES IN THE TRANSITION TO ADA IN THE DON

12 PERSONAL AUTHOR(S)
Shkapsky, Jean M.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1991, September	87

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Ada; Ada Training; Ada Costs

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The Department of Defense has been continually plagued with problems in software development in terms of cost, reliability and performance. To combat these problems, Congress enacted Public Law 101-511, requiring that after June 1, 1991, all Department of Defense software be written in the programming language Ada. However, for this transition to be effective, training of personnel must be accomplished. This thesis addresses issues involved in training of personnel in the Department of the Navy in Ada, the philosophy of training, the number of personnel to be trained and the potential costs involved.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL: LCDR Roger Stemp	22b TELEPHONE (Include Area Code): (408) 646-3113	22c OFFICE SYMBOL: Code OR/St

DD Form 1473, JUN 86 — Previous editions are obsolete. S/N 0102-LF-014-6603

Approved for public release; distribution is unlimited

Analysis of Training-Related Issues
in the Transition to Ada in the DON

by

Jean Marie Shkapsky
Lieutenant Commander, United States Naval Reserve
B.S., University of Missouri-Columbia, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

The Department of Defense has been continually plagued with problems in software development in terms of cost, reliability and performance. To combat these problems, Congress enacted Public Law 101-511, requiring that after June 1, 1991, all Department of Defense software be written in the programming language Ada. However, for this transition to be effective, training of personnel must be accomplished. This thesis addresses issues involved in training of personnel in the Department of the Navy in Ada, the philosophy of training, the number of personnel to be trained and the potential costs involved.

TABLE OF CONTENTS

I. INTRODUCTION

A. DOD PLAGUED WITH SKYROCKETING SOFTWARE COSTS

The Department of Defense (DOD) has substituted the strategy of developing highly-capable electronic systems rather than increasing the numbers of weapons in order to maintain the global balance of power. Unfortunately, this investment in computer technology has not realized its full benefit due to problems in the development of computer software. The complexity of computer systems has continually increased and has left DOD with the following problems (Subcommittee on Investigations and Oversight, 1989):

- software bought or developed does not achieve capabilities contracted for;
- software is not delivered at the time specified;
- software cost is significantly greater than anticipated.

Soaring costs of software is not a new problem facing DOD. As early as 1973, DOD began investigating their ability to combat this phenomenon. This led to the development of the programming language Ada and its adoption in 1980 as an approved DOD high order language (HOL). DOD continued to move in a direction of making Ada not only an approved HOL, but the "standard" HOL. In 1987, DOD Directive (DODD) 3405.2 (canceled February 23, 1991) was published mandating the use of Ada for software development in Mission Critical Computer

Resources (MCCR). DODD 3405.2 required that both a contractor and the in-house development team must obtain a waiver when not using Ada. DODD 3405.1, published immediately thereafter, served to recommend Ada as the standard HOL for automated information systems (AIS), the Navy's business systems. No waiver was required for not adhering to this recommendation.

Although Ada was mandated in DODD 3405.2 for MCCR in 1987, waivers were routinely granted whenever the software developers claimed COBOL, Fortran or something else would be more cost-effective. (Anthes, 1991) With a price tag of $30 billion spent on DOD software in FY90 (Kitfield, 1989), Congress became more interested in DOD software development. Also, as the United States faces a severe shortfall of software professionals, it is anticipated that over the next several years, DOD's demand for new software will soon equal the entire amount it currently has in use.

> ...DOD made perhaps its single most important move to combat software shortages when it established Ada as a common software language in 1980. (Kitfield, 1989)

DOD's problem with software development was no longer its own. On November 5, 1990, Public Law 101-511 was enacted and requires that:

> Notwithstanding any other provision of law, after June 1, 1991, where cost effective, all Department of Defense software shall be written in the programming language Ada, in the absence of special exemption by an official designated by the Secretary of Defense.

With the enactment of the law, Congress removed any doubt on the full-scale commitment it expected of DOD in using Ada

for all major software development efforts. Congress had decided to combat DOD's problem of buying affordable, reliable software on time.

B. FORMATION OF AIP TASK FORCE

In September, 1990, the Assistant Secretary of the Navy for Research, Development and Acquisition (ASN(RDA)) tasked the Director, Department of the Navy for Information Resource Management (DIRDONIRM) with production and issuance of an Ada Implementation Plan (AIP). The AIP was to address Navy and Marine Corps (DON) tactical and non-tactical systems (MCCR and AIS). The purpose was to directly assist acquisition/program managers in meeting the challenges of including Ada into new systems development and upgrades. An AIP Task Force was formed, and held its first meeting on October 4, 1990. The task force's target completion date for development of the AIP was April 1991. Appendix A contains the Task Force members as of that first meeting. At this time Public Law 101-511 had not yet been enacted, but it was clear that AIS software development was to come under similar guidelines as MCCR software development. The author of this thesis became a working member of the AIP Task Force in March 1991.

The Ada Implementation Plan which has recently been renamed as the Ada Implementation Guide, is currently in draft format, being staffed and is expected to be issued in October 1991. For clarity purposes, the term AIP is used. While

awaiting further implementation guidance for the Public Law from DOD, an interim policy guidance was signed on June 24, 1991 by the Assistant Secretary of the Navy for Research, Development and Acquisition (ASNRD&A). The interim guidance strongly states that all Department of the Navy components and activities, including contractors, shall use the programming language Ada for all systems and computer software through all phases of the life cycle. Exceptions are few and can be found in the interim guidance (Appendix B).

An estimate of the cost for full transition to Ada in FY91 is $250 million. (AIP Task Force Minutes, 1990)

C. SCOPE AND METHODOLOGY

The primary emphasis of this thesis is Ada training within the Department of the Navy. To conduct Ada training for the 25 major claimants of DON will require more than $130 million throughout the next five years. This $130 million includes only Department of the Navy in-house training for software professionals. Contractor training is excluded and will require additional funds. (AIP Education & Training Plan, 1991-draft)

The research for this thesis involved a literature review of applicable journals, informal interviews and data collection of training requirements. Interviews were conducted over an eight-month period with software support personnel in both the AIS and MCCR communities. The

interviewees were in positions of management, programming and systems analysis and included personnel in the customer organizations, the users. Their experience level varied within these positions. The training requirement data were gathered from the Office of Civilian Personnel and Management (OPCM), the Bureau of Naval Personnel (BUPERS) and Headquarters, U.S. Marine Corps.

D. THESIS ORGANIZATION

This thesis begins with a discussion of the history of the development of the programming language Ada (Chapter II). DOD has been plagued with skyrocketing software costs and has turned to Ada to help curb these costs. Ada manages concurrent processing, prevents operations on incompatible data, provides modular structure among program components, promotes reusability and is intended for a relatively long operational life thereby lowering maintenance costs. The law mandating Ada has endorsed a new philosophy of a single, transportable, standard support environment of software engineering. Ada is intended to be a tool for this purpose, but is not a cure-all. A recent study suggests that

> ...the use of Ada can be a major--possibly essential-contributor to improving the development and maintenance of software, but it will in no way "solve" all of the problems that plague the DOD in applying computer-based technology. (Emery, McCaffrey, 1991)

Chapter III is an analysis of training and education in the Department of the Navy and focuses on the following

questions: Is education of the benefits of Ada taking place? What is the status of acceptance of Ada in the Department of the Navy and civilian institutions? Has Ada been successfully implemented at the Naval Postgraduate School and the Naval Academy?

Chapter IV discusses the group dynamics of the Task Force. It begins with the origin of the direction of the Task Force, the original format for the AIP and continues through the final meeting in June 1991. The resultant Training Plan not only became an integral part of the AIP, but also will be issued as a stand-alone document.

Chapter V discusses the cost categories of training for each category of programmers/analysts, managers, engineers, support personnel and trainers. A recommended training matrix is provided. A breakdown of the number of prospective Ada trained personnel for Department of the Navy and the overall cost for this training is also provided.

In conclusion, Chapter VI gives recommendations about the future of Ada within the Department of the Navy. The course of a single high order language has been plotted by Congress. However, the success of Ada, and more importantly, software development lies in the hands of the programmers, analysts, managers, and trainers.

II. EVOLUTION OF ADA

A. BACKGROUND

In the early 1970's, DOD experienced a trend of software costs exceeding hardware costs for development of major defense systems. (Boehm, 1973) In 1973, software was 46% (more than $3 billion) of the estimated total DOD computer costs of $7.5 billion. Embedded computer systems comprised 56% of these software costs due largely to their complexity and size. (Fisher, 1979)

It was estimated that at least 450 general purpose languages existed for DOD systems. Depending on the source cited, the actual number varied from 500 to 1500 of high order languages, assembly languages and language variance were considered. No single point of control for each language existed. Therefore, each project office was virtually free to create its own language or use an incompatible dialect of an existing language. The result: diluted training efforts, virtually no technology transfer among projects and a general diffusion of resources. (Booch, 1986)

Since the majority of software costs in DOD were associated with embedded computer systems, DOD directed its attention to embedded systems. A suitable high order language did not yet exist that met the requirements for embedded systems. Embedded applications normally contain thousands to

millions of lines of code and have a typical life span from 10-15 years. They change continuously due to dynamically changing requirements and must be highly reliable. Embedded systems are also typically subject to physical constraints due to target hardware, time and space.

B. DEVELOPMENT OF ADA

In 1975, the joint service High Order Language Working Group (HOLWG) was established. The HOLWG was chartered to: identify requirements for DOD high order languages, evaluate existing languages against these requirements and recommend the adoption/implementation of a minimal set of programming languages. The HOLWG solicited input from all military departments, federal agencies, industry, the academic community and experts from the European computing community. These responses led to a complete set of requirements, representing the desired characteristics for a DOD high order language. Thorough examination found that none of the existing languages fulfilled these requirements. (Whitaker, 1978)

In April 1977, a Request for Proposal (RFP) was issued internationally soliciting designs for the new common high order language, DOD-1. Four contractors were chosen to continue development over a six-month period. Then the field was narrowed down to two finalists. The four original proposals had been color-coded in order to keep ensure that

the reviewers were unaware of the proposal's source. After two public design review meetings, the winner was chosen in May 1979. The Green language became officially known as Ada, the DOD's common high order language. The name, Ada, was in honor of Augusta Ada Byron, Countess of Lovelace, and daughter of the poet Lord Byron and considered the world's first programmer. (Booch, 1986)

The preliminary language reference manual was made public and was also sent to more than 2000 selected experts for their comments. In addition, a public test and evaluation conference was held. Ada had successfully incorporated the particular programming requirements of embedded systems:

- parallel processing;
- real-time control;
- exception handling;
- unique I/O control.

In December 1980, approval was granted for establishing MIL-STD 1815 as the approved DOD standard for Ada. (The number 1815 was chosen since it was the year Augusta Ada Lovelace was born.)

Ada was later standardized and approved by the American National Standards Institute (ANSI) and the International Standards Organization (ISO). (Skansholm, 1988) The government continued its support of Ada by requiring that an Ada compiler must pass over 2000 tests that check for conformance with the ANSI standard. Thousands of computer

scientists took part in the development of Ada and it has proven to be a powerful and consistent vehicle for the efficient creation of software systems.

C. ACCEPTANCE AND FUTURE OF ADA

After almost 11 years, Ada usage is finally expanding significantly. The reaction to Ada has ranged from fierce resistance to simple noncompliance of directives. However, considering it took more than 15 years to become widely accepted for COBOL, another DOD sponsored language, 11 years is not unusual. Early criticisms of both languages included inadequate tools and compilers. Compilers, now conform to the ANSI standard, and development tools have improved, thus absorbing many of the complaints offered by Ada critics. (Anthes, 1991) Ada 9X is a new version of Ada due for release in 1993 and will include functions specifically for business/AIS such as:

- accepting binary-coded decimal data format;
- handling large data base manipulation;
- supporting the 64-bit fixed-point arithmetic.

Listed as a study topic for inclusion in Ada 9X is support for object-oriented programming (OOP). The proposed support of OOP concepts would adopt the qualities of inheritance and polymorphism. Object-oriented programming is particularly useful for evolutionary programming and would further enhance

Ada's ability to interface with other resources and software/ code reusability.

The impact of Ada can be seen by the monetary expense. According to Focused Ada Research Corporation, in 1989 users spent $144 million on Ada software products, bought or used $831 million in hardware for Ada development and paid an additional $1 billion in direct salaries to Ada programmers. They estimated the value of Ada-based systems development projects ran in the tens of billions of dollars. However, as difficult as it is to measure DOD use of Ada, commercial use of Ada is even more difficult because users tend to guard their success stories as closely as trade secrets. (Anthes, 1991)

Congress has mandated that Ada will be adopted as DOD's standard programming language by enacting Public Law 101-511. DOD led the development of Ada with the hope that a single language would allow development of reusable code thus freeing scarce programmer resources to concentrate their development efforts on the unique software requirements of each new system. The strong software engineering discipline that Ada supports increases the level of attention on front-end requirements. Software development with Ada encourages a complete systems analysis approach and therefore life cycle considerations are an important aspect of each decision making process.

Public Law 101-511 has put high visibility on the choice of programming languages used for system development. Commands vying for funds are aware that non-compliance of Ada directives is a sure way for their programs to get "axed" from the budget. The Department of the Navy commands may request waivers through Commander, Navy Information Systems Management Center (NISMC), but they most likely will not be approved.

III. EDUCATION AND TRAINING OF ADA WITHIN DON

A. EDUCATION AND TRAINING OVERVIEW

The Armed Services have been traditionally known for outstanding training in their warfare specialties. Very few individuals are recruited pre-trained as "machine-gunners," "ship-drivers," or "jet pilots." With the split-second timing required in combat, many specialties are taught to "react," not to debate questions of "Should I?" or "Shouldn't I?". However, not only has training of DOD software professionals been traditionally poor, but DOD primarily selects program managers from those military officers whose career paths have reached a stage at which they are ready for large scale project management. Technical expertise in the respective project area is usually a secondary consideration. Furthermore, the difficulty in finding civil service personnel who are properly trained and who are also talented program managers has created a "quiet crisis" within DOD. (Subcommittee on Investigations and Oversight, 1989)

The Department of Defense should not bare the entire responsibility for this shortfall since nonavailability of trained personnel, cost overruns, reliability and performance problems with software systems plague private industry as well. With the prediction of future shortages of software professionals due to increase demands for new software

(Kitfield, 1989), DOD may find it even more difficult to attract the "best and brightest" members within the field. This predicted deficit is due primarily to DOD's inability to offer starting salaries that are competitive with those offered in private industry. (Subcommittee on Investigations and Oversight, 1989)

There is an important distinction between education and training.

> Education involves an understanding of abstract theory; training involves gaining the skills necessary to accomplish a task. Without adequate training, users will not have the knowledge to use the technology to its maximum benefit. (Mensching and Adams, 1991)

However, the Department of the Navy has failed in educating its personnel in the advantages that can be gained by using Ada in conjunction with sound software engineering concepts and in training its personnel in the principles of software engineering. (Knight, 1990) Even within the AIP Task Force, representatives of both the AIS and MCCR communities had not previously been educated in the benefits of software engineering complemented with Ada. This general lack of education in the area of software engineering must be overcome before training can ever achieve its full benefit. Software professionals need to be made aware that properly applied software engineering principals coupled with the programming power Ada has to offer, can lead to increased programming productivity. Productivity can be significantly increased because of the relative ease in which Ada program components

can be integrated, a reduction in program maintenance and an ability to reuse previously tested and validated code.

B. DON ACADEMIC INSTITUTIONS

The Department of the Navy's primary academic institutions have been slow to take the initiative in this arena. Therefore, it is no wonder civilian academic institutions have not been quick to incorporate Ada into their curricula. The Naval Postgraduate School (NPS) has been teaching Ada as its primary programming language since March 1989. However, the predominant philosophy has been that teaching Ada is no different than teaching other programming languages. It would be more effective to accompany the instruction of ADA together with the basics of software engineering. Otherwise, teaching ADA just as another programming language would be insufficient to introduce the concept of software engineering to its officers who may, one day, be program managers. Ada is not an easy language to learn and requires more experience than other languages before personnel can become proficient. (IIT, 1989) Therefore, by not teaching Ada in its full context, not only does the Department of the Navy miss an opportunity, it may actually have negative repercussions by "souring" its future program managers with such a difficult language.

Although NPS offers Ada as a primary programming language, it is required for only two of the approximately 40 curricula: Computer Science and Information Technology Management. Of

the remaining 32 curricula, approximately 70% are considered technically oriented. Approximately 800-900 students a year graduate from NPS having absolutely no required contact with Ada. From a quick check of potential billets available, approximately 20% of these personnel will be future program managers for the Department of the Navy.

The Naval Academy is in the process of revising its curriculum on Ada. Ada was previously taught as a first language at the Naval Academy, but was dropped from the curriculum because it was "too difficult." A recent article published by two instructors at the Naval Academy may account for this decision.

> The fundamental problem is found in the power of Ada. When constrained to the narrow confines of a simple classroom example, it can often inhibit the learning process. The language is a powerful tool that, in the hands of an expert, produces well-designed, elegant solutions. The languages's features, however, can overwhelm the average student struggling to produce a 50 line program. (Spegele, Park, 1991)

Ada is a robust language and adds a level of complexity which can often impair learning for the novice. However, what kind of a message is the Department of the Navy sending to private industry, to vendors and to its own commands when their own academic institutions cannot solve these issues?

C. EDUCATION AS A LONG-TERM INVESTMENT

Proper education is the key for achieving the long-term benefits which can be gained through the use of Ada. Most students in civilian academic institutions are not yet taught

Ada in a software engineering environment context. Rather, they are just taught the mechanics of coding. (Subcommittee on Investigations and Oversight, 1989)

Education and training are the keys to making the transition to the "Ada mindset."

> The mindset involves learning and applying new software engineering principles, modern methods like OOD (Object oriented design), and advanced packaging concepts and tools, as well as the programming language itself. (Reifer, 1991)

The emphasis here is on changing the way business is currently being done by looking at the "whole picture" in a software engineering sense. Making this change will place additional requirements on the education and training process. However, these requirements are minimal and the net payoff will be well worth the investment made.

IV. AIP TASK FORCE GROUP DYNAMICS

A. ESTABLISHMENT OF THE AIP TASK FORCE

The AIP Task Force was chaired by a member of the DASN(IRM) staff, with a deputy chair from the Space and Naval Warfare Systems Command (SPAWAR). SPAWAR became involved because they had been in the process of drafting an AIP for the MCCR community. This AIP had previously been required under SECNAVINST 5234.2 (canceled by DODD 5000.1). Since much of the outline for the SPAWAR AIP had been completed, it was used as the base document. This may have been the cause of later discussions within the Task Force that the AIP was heavily weighted towards the MCCR community.

B. BUILDING THE AIP

The first meeting of the Task Force was held on October 4, 1990, at SPAWAR in Arlington, VA. Appendix C is the initial outline for the AIP which was presented at that meeting (a section on education and training was not included initially). The Task Force began with 17 members from various command backgrounds, some of whom were sold on Ada and others who were skeptical. Many of the members had been previously assigned to specific groups by the chairperson; however, those in attendance who were not previously assigned a specific work group were assigned at the meeting.

Many of the members had been seeking guidance on Ada policy and were anxious to comply, but had been overridden by managers who did not understand the long-range benefits Ada could offer in the areas of software acquisition and development. All members realized, however, that Ada is here to stay and with that knowledge alone, their respective commands would benefit.

The purpose for the AIP was to describe a strategy for successful use of Ada and software engineering in the Department of the Navy for both MCCR and AIS acquisitions. The style was pre-selected to have a handbook flavor for ease of use by the Program Manager at the Systems Command level.

Work continued on the expansion of the AIP. By late October of 1990, the Task Force was aware that the House Appropriations Committee (HAC) had proposed a public law to be effective June 1, 1991, which would mandate the use of Ada for all MCCR and AIS software developments. DASN(IRM) had requested the Task Force assist in preparing three point papers: the first, addressing implementation of the law; the second, addressing the waiver or exception process; and the third, the impact upon the Department of the Navy by accelerating the current program to meet the June 1991 date.

The Task Force would fully support the HAC bill, but in the point papers they advocated a phased approach to transition to Ada over the next ten years. Training was addressed as a major impact area. It was noted that due to

compliance with previous directives, the MCCR community was significantly ahead of the AIF community in transiting to Ada. However, both the AIS and MCCR communities were a long way from full implementation, partly due to budget and hiring constraints. No additional money had been programmed for this transition and a portion of the previously approved funding had been deducted from the budgets for IRM due to the Corporate Information Management (CIM) initiative. CIM was consolidating ADP/IRM functions under one roof for DOD and the amount which had been deducted was the anticipated savings that the consolidation was to reap for DOD.

By February 1991, with the enactment of Public Law 101-511, the purpose of the AIP had changed. The AIP was now directed at providing guidance to project managers and their staffs on implementing Department of the Navy policies and standards for use of the Ada programming language. An updated outline is provided in Appendix D.

The final formal meeting of the AIP Task Force took place June 11-13, 1991, with a membership count of 37. (See Appendix F.) The page count of the AIP had grown proportionately with the number of personnel added to the Task Force. Copies of the AIP had previously been sent to members of the Task Force for their comments and returned for reproduction prior to this meeting. Section groups were divided up into separate small groups for reviewing comments and generating mark-ups of the AIP.

Many members of the Task Force were disappointed that the AIP had become more of a Guide for Implementing Ada, vice a plan. During discussions concerning the Air Force's Ada Implementation Plan of January 29, 1989, which simply stated policy, the suggestion was made to take out Section 2.0 on DON policy and issue it as a separate instruction which referenced the "Guide" for assistance. By June 1991, the new title of "Department of the Navy Ada Implementation Guide" was given to the entire document. After further review, the chairperson of the Task Force agreed that there should be a brief plan, similar to the Air Force AIP, stating the Department of the Navy policy. The Ada Implementation Guide would still provide assistance to the program manager, but the policy would be stated in the instruction.

A draft instruction was prepared which was signed later in June by DASN(C4I/EW/Space). This instruction became the Interim Department of the Navy Policy on Ada (see Appendix B). DASN(C4I/EW/Space) believed this would be a more effective approach in meeting the June 1, 1991 deadline established by Public Law 101-511. The Ada Implementation Guide is expected to be issued in October 1991.

C. INITIATION OF EDUCATION AND TRAINING PLAN

Training was initially listed as a subheading buried deep under Ada Related Issues in an appendix. In December 1990, it was decided that a separate appendix was to be added on DON

training requirements. In February 1991, an outline of the Training Plan, shown in Appendix E, was presented. The strategy behind the outline was that an actual training plan was needed to address the Department of the Navy's infrastructure training vice a guide for developing that plan. A representative from the Naval Postgraduate School was added to the training section to research Ada training sources and the costs associated with that training. The Training Plan came under severe scrutiny because it was intended to be published not only as an appendix, but also as a stand-alone document. Discussion continually arose concerning the value of Ada over other programming languages. Was training a programmer in Ada any different than training a programmer in COBOL, Fortran or any other language? The purpose of the AIP was not to convince anyone to use Ada, that came from Public Law 101-511. Rather, it was to emphasize that good software and systems engineering practices are the keys to a successful program. DOD now has a standard programming language which supports software engineering and in order to reap the rewards, proper training is required in areas other than simple programming.

D. PRESENTATION OF THE EDUCATION AND TRAINING PLAN

The Training Plan had expanded, but the DASN(C4I/EW/Space) staff now wanted more detailed statistics for use in future Program Objective Memorandum (POM) cycles. This required a

breakdown of personnel within DON who needed Ada training by organization along with the overall cost of this training. The author began gathering data on the number of DON personnel potentially needing Ada training and worked with Naval Computer and Telecommunications Station, New Orleans representatives on developing a complete cost analysis for the Training Plan.

By June, 1991 the general consensus was that the Training Plan now contained too many DON statistics which would only serve to confuse project managers. However, in order for the Training Plan to be effectively used as a stand-alone document as well as provide useful input for POM cycles, the DASN(C4I/EW/Space) chairperson insisted that they remain a part of the document. The number of DON civilians requiring Ada training was believed to be low in the MCCR community. Members noted that virtually every civil service specialty series working in the MCCR community would require some type of Ada training. Further research continued on identifying additional civil service specialty series training requirements, after which the statistics were recomputed. Chapter V provides the details of the process used in identifying these requirements and how estimated training costs were obtained.

V. COST ANALYSIS AND CATEGORIES OF TRAINING

A. DESCRIPTION OF PROBLEM

The Naval Computer and Telecommunications Command (NCTC) requested a study on the impact of implementing the Ada programming language at the eight Naval Regional Data Automation Centers (NARDACS). NCTC is a central design agency which invests heavily each year in software development and was one of the 25 major claimants used in the study. (A complete list is shown in Figure 1.) Of the 1020 programmers on board the NARDACS, only 31 programmers had received Ada training as of the end of FY90. Of those 31 programmers, 22 had received only a one-week course and had not yet received practical experience in Ada. This study included in-house contractors as well as DON software support personnel. (Knight, 1990)

After conducting interviews with several other commands, the author found this not to be unusual on the AIS side of the Department of the Navy. The MCCR side was found to be somewhat better, probably because Ada had been mandated since 1983.

Even fewer personnel are experienced to date in software engineering using Ada. Without additional training in software engineering, the Department of the Navy will lose many of the benefits Ada has to offer.

Chief, Bureau of Medicine and Surgery
Chief of Naval Education and Training
Chief of Naval Operations
Commander-In-Chief, U.S. Atlantic Fleet
Commander-In-Chief, U.S. Naval Forces, Europe
Commander-In-Chief, U.S. Pacific Fleet
Commander Naval Reserve Forces
Immediate Office of the Secretary
U.S. Marine Corps
Military Sealift Command
Naval Air Systems Command
Naval Computer and Telecommunications Command
Naval Facilities Engineering Command
Naval Intelligence Command
Naval Military Personnel Command
Naval Oceanography Command
Naval Sea Systems Command
Naval Security Group Command
Naval Special Warfare Command
Naval Supply Systems Command
Navy Field Offices
Navy Staff Offices
Office Chief of Naval Research
Space and Naval Warfare Systems Command
Special Programs Office

Figure 1. Major Claimants

Ada simply provides many facilities and mechanisms which can be used to support portability. The design of the underlying software system provides the portability of the systems, not the language which it is implemented. (Engle, 1991)

Successful implementation of Public Law 101-511 requires establishment of a Department of the Navy education and training program designed to generate sufficient numbers of personnel proficient in software engineering using Ada. However, to date, no research has addressed the issue of how many software support personnel there are within the Department of the Navy or what the cost of training those personnel would be. The following questions needed to be answered:

- What personnel need to be trained in software engineering using Ada?
- How many personnel will require the training?
- What will the cost of this training be over a five-year period?

Note: A five-year period was selected for budgeting purposes with the DON Program Objective Memorandum (POM) cycle.

B. METHODOLOGY

Prior to this author's participation with the Task Force, prior research had broken DON software professionals into five categories: managers, engineers, programmers/analysts, project support personnel and trainers. The following descriptions of each of these categories were extracted from the Ada Implementation Plan (AIP, 1991, draft).

1. Manager

Top and middle managers are defined as those responsible for high-level planning and decision making in organizations. They need awareness and orientation training on the benefits, capabilities, and differences of software

engineering using Ada so that they can provide planning, direction, and support for Ada implementation.

Project managers are defined as those responsible for software projects. Usually, these managers select people for specific assignments, choose equipment and software tools, estimate costs, and plan schedules. Therefore, they need orientation and project management training on software engineering using Ada so that they can make informed technical decisions, develop plans, and conduct evaluations. Failure to understand the unique aspects of Ada will cause mismanagement and excessive cost in systems development and post deployment support.

2. Engineers

Defined as those responsible for system engineering and top-level design, engineers usually interface with project managers and programmers and are responsible for all or major components of systems. They need orientation, software engineering, programming, development environment, and quality assurance training in software engineering using Ada. Many engineers may need only fundamental, not advanced, training in Ada programming; the need is dependent on the individual project and the interaction between the engineers and programmers.

3. Programmers and/or Analysts

Programmers and/or analysts, defined as those who program and test computer programs, initially need orientation, software engineering, and programming training in software engineering using Ada. Later, they need training in Ada development environments and project management. Programmers and/or analysts with backgrounds in Pascal and other High Order Languages (HOL) incorporating systems engineering principles should adapt to and progress faster in Ada training than programmers and/or analysts with a strong background in languages such as COBOL and FORTRAN.

4. Project Support Personnel

Project support personnel are technical and nontechnical personnel who provide administrative support in contracts, purchasing, and budgeting or who deal with configuration management, quality assurance, technical documentation, libraries or data management control, partitioning, and integration. Project support personnel usually interact with project managers and systems engineers. They need training in the fundamentals of software engineering using Ada, particularly in the way it

differs from other HOLs (e.g., coding style, library structure).

5. Trainers

Educators provide training support by establishing training plans, course evaluation, procurement, arrangement, preparation, instruction, and maintenance of training records. Training personnel usually have experience as administrators or instructors and interact with project managers. Trainers performing planning and administrative functions need an orientation to and understanding of the fundamentals of software engineering using Ada. Trainers preparing and performing Ada technical instruction need full exposure to and experience with Ada.

The Education and Training group conducted interviews with various organizations on both the MCCR and AIS side and drew from their own experiences at NARDAC San Francisco and NCTS New Orleans. The author continued with those interviews, conducted further literature review and gathered additional numeric data. The numbers of software support personnel were gathered from the following data bases: OPCM, BUPERS and Headquarters, USMC and was correct as of April 30, 1991.

C. COST ANALYSIS

In seeking the number of personnel requiring Ada training, the five categories first needed to be broken into civil service specialty series and military specialties. Through a series of interviews and cooperative effort with NCTS New Orleans, the author broke down the categories into the following series and specialties.

1. Civilian Series

- 0334: Computer Specialist;
- 0854: Computer Engineer;
- 1515: Operations Research Specialist;
- 1550: Computer Scientist.

2. Military Personnel

- Navy: officers;
  - -- Subspecialty code Description
  - -- 0095P/0095Q Computer Information Manager;
  - -- 0091P/0091Q Computer Technology;
  - -- 0090P/0090Q Hold both of above.
- Navy: enlisted (specifically DPs);
  - -- NEC Description
  - -- 2741 Programmer/Assembler;
  - -- 2742 Programmer/COBOL;
  - -- 2743 Programmer/Fortran;
  - -- 2751 Systems Analyst.
- USMC: officers;
  - -- MOS Description
  - -- 4002 Data Systems.
- USMC: enlisted;
  - -- MOS Description
  - -- 614 Programmer/COBOL;
  - -- 55 Programmer/Ada*.

* can only be given as a secondary MOS (personnel must first hold MOS 614)

Within the Department of the Navy these specialties totaled to 14,091 software support personnel located in the AIS and MCCR communities. Software support personnel are broken down as follows:

- 11,947 Civilians (Civil Service employees);
- 268 U.S. Marine Corps Officers;
- 614 U.S. Marine Corps Enlisted Personnel;
- 455 U.S. Naval Officers;
- 807 U.S. Naval Enlisted Personnel.

Of these 14,091 personnel, not all would require Ada training since current Department of the Navy policy (Appendix B) does not require Ada for smaller software development (i.e., cost less than $50K in development and $5K/yr in maintenance). After reviewing previous studies of past and projected software development, the group came to the general consensus to include 50% of all personnel and an additional 10% to account for personnel turnover. Using these percentages a formula for establishing a baseline figure for Ada training was established.

Baseline personnel to be trained = .5P + .10T (1)

where:

P = total software support personnel,

T = .5P.

Each of the five categories of personnel was computed separately and totaled using Equation (1). From these computations, it was determined that a baseline of 7750 personnel needed to be trained over the next five-year period.

NCTS New Orleans had been investigating all Ada training currently available and had estimated an average cost of $200/day for individual training. This average cost was the constant used in the cost analysis. Table 1 represents the overall training costs based on the recommended training matrix (Figure 2). The initial conclusion was that a total of $57 million over the next five years would be needed to implement the proposed Department of the Navy training plan necessary to achieve full scale implementation of Ada.

TABLE 1

ADA TRAINING COSTS

	FY92	FY93	FY94	FY95	FY96	TOTALS
	(dollars in millions)					
Manager	.4284	1.2750	1.4926	.6392	.4284	4.2636
Engineer	.3616	1.0880	1.2672	.5440	.3616	3.6224
Programmer	4.8480	14.5536	16.9824	7.2678	4.8480	48.5088
Support	.0512	.1536	.1824	.0800	.0512	.5216
Trainer	.0510	.1496	.1734	.0748	.0510	.4998
TOTALS	5.7402	17.2330	20.0980	8.6148	5.7402	57.4162

ORIENTATION COURSES	LENGTH	AUDIENCE* MNGR	ENGR	PGMR	SUPP	TRNR
Ada Overview	2 Hours	x	x	x	x	x
Ada for Executives	7 Hours	x				x
Ada for Software Managers	7 Hours	x				x
Ada for Engineers/Programmers	7 Hours		x		x	x
Ada Acquisition Planning	7 Hours	x	x		x	x
SOFTWARE ENGINEERING COURSES						
Ada Software Engineering	3 Days	x	x		x	x
PROGRAMMING COURSES						
Ada MCCR Programming	5-10 Days			x		x
Ada AIS Programming	5-10 Days			x		x
Advanced Language Concept	[need length]		x	x		
Ada as a First Language	10-15 Days			x		x
Ada Refresher Programming	5 Days			x		x
Ada Data Structures	5 Days			x		x
Ada Tasking	5-10 Days			x		x
Ada Project Experience	Varies	x	x	x	x	x
DEVELOPMENT ENVIRONMENT COURSES						
Ada Program Support Environment	2-3 Days	x	x	x	x	x
Ada Run-Time Environment	2-3 Days	x	x	x	x	x
PROJECT MANAGEMENT COURSES						
Ada Project Management/ Ada Cost Estimating	2-3 Days	x	x	x	x	x
Ada Contracting	2-3 Days	x	x	x	x	x

* Legend
MNGR = Manager
ENGR = Engineer
PGMR = Programmer and/or Analyst
SUPP = Support Personnel
TRNR = Trainer

Figure 2. Recommended Training Matrix

However, as discussed in Chapter IV, when these data were presented to the Task Force in June 1991, personnel from the MCCR community found certain assumptions to be inaccurate. Specifically, they believed there were other civil service specialty series involved with Ada and that a much higher percentage of all software support personnel would require training.

Through additional interviews, the Education and Training group discovered these personnel had a valid argument. Within the MCCR community, there was a much higher percentage of personnel that are and would be directly involved with Ada. The following additional civil service specialty series were added to the study:

- 0855 Electronic Engineer;
- 1520 Mathematician;
- 1300 Physicist;
- 0510 Accountant.

However, only those personnel with a civil service grade of GS-12 and above in these additional series were added. Most of these personnel fell in the category of managers with a much broader scope of responsibility than their series may indicate. Additionally, it was felt that more than 90% of all MCCR software support personnel would require some sort of training in Ada. However, the 10% turnover factor was still considered to be a valid assumption.

Therefore, for the MCCR community, the formula used for estimating the baseline number of personnel to be trained was revised as indicated in Equation 2.

$$\text{Baseline MCCR personnel} = .9P + .10T \qquad (2)$$

where:

P = total MCCR software support personnel,

T = .9P.

Upon further review, it was felt that Equation (1) was still valid for determine baseline training needs for AIS software support personnel. By including the additional civil service specialty series, the total number of DON software support personnel was estimated to be 26,929. This total included 11,850 additional personnel from the MCCR community and 988 from the AIS community. Recomputing using the revised MCCR formula, the total baseline figure for personnel was estimated to be 22,855.

Table 2 is a breakdown of the training costs by categories over a five-year period and includes the total cost for training within each category. The total revised cost for training the baseline number of personnel in Ada, as shown in Table 2, is $130 million and was considered a reasonably accurate estimate by DASN (C4I/EW/Space).

TABLE 2

REVISED ADA TRAINING COSTS

	FY92	FY93	FY94	FY95	FY96	TOTALS
	(dollars in millions)					
Manager	1.7536	5.2640	6.1440	2.6336	1.7536	17.5488
Engineer	2.1270	6.3750	7.4400	3.1890	2.1270	21.2580
Programmer	7.4880	22.4448	26.1792	11.2224	7.4880	74.8224
Support	.3900	1.1730	1.3680	.5850	.3900	3.9060
Trainer	1.2852	3.8556	4.4928	1.9224	1.2852	12.8412
TOTALS	13.0438	39.1124	45.6240	19.5524	13.0438	130.3764

VI. RECOMMENDATIONS AND CONCLUSIONS

A. RECOMMENDATIONS

The current low acceptance rate of Ada within the Department of the Navy is due to the lack of a formal education and training program. This exists in spite of solid evidence that Ada has largely achieved its goal of providing a first-rate development environment for very large systems. (Emery, McCaffrey, 1991)

A training matrix containing an average Ada curriculum for the five categories of software professionals was shown in Figure 2. It is a comprehensive list of courses, which are needed by most personnel, and was developed from training experiences and suggestions of the members of the AIP Task Force. However, project managers/training planners at each activity or for each project should conduct their own training needs analysis. The Project Manager (PM) first evaluates the current skill level of the work force on the project and then determines the skills required for the projected system environment. By comparing the two skill levels the Project Manager will have identified specific capability gaps. (U.S. General Services Administration, 1990) Finally, by using the matrix shown in Figure 2, the Project Manager should be able to realistically define the additional training required.

Training should be given precedence in the budgeting process. Federal funds should be provided for the development and dissemination of teaching methodologies which emphasize both software engineering and Ada. Encouraging civilian academic institutions will not only provide a broader base of software professionals for DON/DOD to choose from, but will also serve to reduce the projected shortages of software professionals. In addition, with more professionals trained in solid software engineering principals, code reusability will become more commonplace, thus also reducing the overall software demand.

Code reusability, however, cannot be maximized without providing a greater flexibility in the software acquisition policies under which the Project Manager must operate. No royalties or compensation are offered to software developers for software reuse. Furthermore, DOD refuses to relax their policy on requiring complete data rights packages. The front end costs associated with building reusable code are high and many private industries are not willing to participate in low-bid contract competition knowing that their software will be included in a common DOD software library without future royalty considerations. (Kitfield, 1989) Top level acquisition managers must be educated in the long-term benefits of software engineering and a more flexible policy provided for Project Managers.

This short-term mentality must be overcome and long-term solutions put into effect. The cost of transition to Ada is no small matter in DON or in private industry.

> The traditional short-term financial orientation of U.S. firms works against the adoption of Ada and its attendant software engineering disciplines. Getting into Ada may cost hundreds of thousands of dollars in software and more in training, according to industry analysts. The savings in reusable code and reduced software maintenance may be huge, but might not show up for years. (Anthes, 1991)

Kurt Lewin describes the process of bringing about effective change as a three-step process: unfreezing, changing, and refreezing (Lewin, 1947). Chapter III discussed education and the Department of the Navy's failure to make this change obvious by educating its personnel not only in software engineering with Ada, but also with an appreciation of the problem. Mid-level managers must take on the burden of most of this "awareness-type" education. They must not assume that their personnel fully understand the problem or comprehend the full benefits which can be realized through full Ada implementation. Most often the personnel "in the trenches" are only concerned that their programs are valid and function according to specifications.

> Few of the development sites actually understand or employ software engineering principles. Therefore, touting Ada as supporting software engineering means nothing to the programmers in the trenches. And without convincing the "techies, any transition effort will be torpedoed. (Knight, 1990)

The House Appropriations Committee has acted as the change agent by enacting Public Law 101-511. However, with the

exception of the Interim Policy Guidance, very little has been done to assist in this change. The Corporate Information Management program under DOD has yet to issue any formal guidance on Ada. Department of the Navy commands must take a proactive approach to Ada. This will assist in the refreezing aspect of the change. There is strong opposition to Ada from many personnel, largely due to their inability to see the change in a positive light. Managers must look to the future. A loss of one or two personnel who refuse to accept the transition may cause an immediate drop in productivity, but may be a reality as managers see more existing and new development in Ada.

B. CONCLUSIONS

In order to ensure that the Department of the Navy will reap the reward of reliable, transportable, cost-effective software systems, we must train our personnel in project management and solid software engineering practices using Ada.

Public Law 101-511 has set the course by mandating Ada. A standard has been set and should not be softened. Cost-effective, reliable software is achievable using software engineering with Ada and Department of the Navy should not be influenced by personnel who are unwilling to accept change. This is a long-term program and until metrics are available that can show that the premise of cost savings <u>cannot</u> be realized using Ada, strict adherence should be required.

Future research will be necessary and a cost-benefit analysis conducted as solid data becomes available.

> And the Lord said, Behold, the people is one, and they have all one language; and this they begin to do; and now nothing will be restrained from them, which they have imagined to do.
>
> Genesis 11:6
> King James Version

The Department of Defense has adopted one standard language, Ada ANSI/MIL-STD-1815A-1983, which has been repeatedly criticized for its limitations. However, by taking full advantage of the inherent features of Ada and the future enhancements proposed for inclusion in the new version of Ada, Ada 9X, the Department of the Navy can make great strides in software development particularly in terms of cost, reliability and performance.

APPENDIX A

TASK FORCE MEMBERS AS OF OCTOBER 4, 1990

DEPARTMENT OF NAVY
ADA IMPLEMENTATION PLAN
TASK FORCE PARTICIPANTS

CHAIR:	Ms. Antoinette Stuart	DASN (IRM)
DEPCHAIR:	CDR Martin Romeo	SPAWAR

AIP Task Force Representatives:

NAVSEA	Gregg Engledove Clive Harding
NAVAIR	Tom Coyle
NCTC	Joan McGarity
NADC	Hank Stuebing
NOSC	Bob Calland Rich Bergman Cathy Ruiz
NSWC	Dan Green Frank Ervin
NUSC	Tom Conrad D. Labossiere
FCDSSA San Diego	George Robertson
FCDDSA Dam Neck	
USMC	Captain G. Despasquale Captain D. Thompson

APPENDIX B

INTERIM DON POLICY ON ADA

This appendix is the interim Department of the Navy policy on Ada implementation. It was issued in June of 1991.

DEPARTMENT OF THE NAVY
OFFICE OF THE ASSISTANT SECRETARY
(Research, Development and Acquisition)
WASHINGTON, D.C. 20350-1000

JUN 24 1991

MEMORANDUM FOR DISTRIBUTION

Subj: INTERIM DEPARTMENT OF THE NAVY POLICY ON Ada

Ref: (a) U.S. Congress. Department of Defense Appropriations Act 1991. Public Law 101-511 (Nov. 5, 1990), 104 Stat. 1856-1914
(b) DODI 5000.2 of 23 Feb 91

Encl: (1) Interim Ada Programming Language Policy

Reference (a) states "notwithstanding any other provision of law after June 1, 1991, where cost effective, all Department of Defense software shall be written in the programming language Ada, in the absence of special exemption by an official designated by the Secretary of Defense".

The office of the Secretary of Defense has not yet provided implementation guidance for this law. Pending receipt of further policy, enclosure (1) is the Department of the Navy interim policy for the use of Ada both in Automated Information System (AIS) and Mission Critical Computer Resources. Please ensure that the intent of the law and interim policy in enclosure (1) are complied with and implemented within your organization.

Reference (b) remains applicable for MCCR and is only reinforced by this interim DON Ada Policy.

It should be fully recognized that this is interim policy. Anticipating that implementing guidance from OSD soon will be available, this policy will remain in effect for six months. During this period, significant difficulties experienced with the policy should be brought to the attention of Commander, Naval Information Systems Management Center (NISMC). Until Commander, NISMC is formally established, correspondence concerning this policy for him will be sent to Deputy Assistant Secretary of the Navy (C4I/EW/Space).

Gerald A. Cann

Distribution:
CNO
CMC
AAUSN
CNR
(See next page)

Subj: INTERIM DEPARTMENT OF THE NAVY POLICY ON Ada

Copy to:
COMNAVAIRSYSCOM
COMSPAWARSYSCOM
COMNAVSEASYSCOM
COMNAVSUPSYSCOM
COMNAVFACENGCOM
COMNAVCOMTELCOM
CINCLANTFLT
CINCPACFLT
CINCUSNAVEUR
COMSECONDFLT
COMTHIRDFLT
COMSIXTHFLT
COMSEVENTHFLT
SSPO
PEO (Air ASW)
PEO (TACAIR)
PEO (CM and UAV)
PEO (Space)
DRPM (AEGIS)
PEO (Submarine Systems)
PEO (Surface ASW)

Subj: INTERIM ADA PROGRAMMING LANGUAGE POLICY

Ref: (a) SECNAVINST 5200.32
(b) SECNAVINST 5231.1()
(c) SECNAVINST 5430.20C
(d) DODD 3405.1 of 2 Apr 1987
(e) DODD 5000.2 of 23 Feb 1991
(f) NBS FIPS Publication 11-2 of 9 May 1983
(g) DOD Standard 2167A of 29 Feb 88

Attachments: (a) Ada Exception Notification Format
(b) Ada Waiver Request Format

1. Purpose. To establish policy for using the programming language Ada in the development and maintenance of software for systems managed under references (a), (b) and (c).

2. Background. Public Law 101-511, Section 8092, requires that after June 1, 1991, where cost effective, all Department of Defense software be written in the programming language Ada. This instruction provides Department of Navy (DON) policy concerning the use of Ada and complies with Department of Defense (DOD) policy contained in references (d) and (e).

3. Definitions. Terms used in this instruction are defined in reference (f), except special terms defined as follows:

a. Ada-Based AST. An AST that specifically supports Ada software development (e.g., Ada source code generator, DBMS with Ada interface, etc.).

b. Advanced Software Technology (AST). Software tools, life-cycle support environments (including program support environments), non-procedural languages (4GLs), modern database management systems (DBMSs), software tools, and other technologies that provide improvements in productivity, useability, maintainability, portability, and other benefits, over those capabilities commonly in use.

c. Commercial-off-the-shelf (COTS) Software. Software (including operating systems, utilities and stand-alone applications programs) already developed, tested, and sold to other DOD or commercial customers, supported by a commercial vendor over the system life cycle, and requiring no government modifications over the system life cycle.

d. DOD-Approved High Order Languages (HOLs). The languages listed in reference (d): Ada, C/ATLAS, COBOL, CMS-2, FORTRAN, JOVIAL, Minimal BASIC, PASCAL, and SPL/1.

e. Exception. An exception is approval to adopt an authorized non-Ada approach contained in this instruction which will require only limited justification and reporting.

f. Fourth Generation Languages (4GLs). Non-procedural computer programming languages which consist of compact, English-like statements which describe the overall tasks a computer is to carry out without specifying any individual steps or their order. For the purpose of this policy, 4GLs include products which generate HOL code.

g. Milestone Decision Authority. The individual designated to approve entry of an acquisition into the next phase in accordance with applicable directives.

h. Rapid Prototype. Quick trial implementation whose main purpose is to assess the feasibility of the product, verify system requirements and then discard.

i. Validated Ada Compiler. A compiler registered with the Ada Joint Program Office (AJPO). A project-validated compiler, a compiler that is registered with the AJPO at project start or Milestone 0, is considered validated for the entire life cycle of the designated project.

j. Waiver. A waiver is approval to deviate from policy contained in this instruction which will require a detailed justification to support.

4. Applicability. This instruction applies to all systems and computer software managed under references (a) through (c), all phases of the life cycles of those systems and software, and all DON components and activities, including their contractors.

5. Scope. This instruction covers all computer software except:

a. Software which has already been operationally fielded and for which maintenance activity is restricted to error correction.

b. Systems that have entered production and deployment or have passed milestone II of references (a) or (b), but have not been operationally fielded as of 1 June 1991.

c. Systems for which a documented language commitment was made in compliance with previous policy.

d. Non-deliverable software as defined in reference (f).

e. Software developed for dedicated processors that have 16-bit or less instruction set architectures and less than 256K total memory.

f. Software for use in projects at a single site and cost less than $50K in development and $5K/yr in maintenance.

g. Software written by individual personal computer/workstation users for personal or intra-office use, for which DON maintenance activity support will not be provided.

6. Policy. It is DON policy to:

a. Use the Ada programming language, as defined in ANSI/MIL-STD-1815A-1983, as the single, common, high order computer programming language for all computer resources. A validated Ada compiler and modern software engineering principles that facilitate the use of Ada must be used, unless a waiver or exception has been approved.

b. Meet DON software requirements, by reusing existing Ada code whenever possible.

c. Grant waivers to the policies in this instruction on a specific system and subsystem basis only. Further, to base the waiver decision on an analysis of total life-cycle costs, impact, and potential for reuse in other DON and/or DOD acquisitions.

d. Identify needed technologies that have the potential to facilitate the use of Ada in future systems acquisitions and to aggressively acquire those technologies.

e. Whenever technically feasible and cost effective, acquire computers for which validated Ada compilers have been developed and to include language to this effect in contractual matters pertaining to all system acquisitions.

f. Use an Ada-based program design language that can be successfully compiled by an Ada compiler, during the design of software to improve the portability of the software design.

g. Use modern software engineering principles and Ada-based ASTs which facilitate the use of Ada in order to reduce costs, shorten schedules, and improve software quality.

7. Exception Categories. For the categories listed below, an exception request that documents a project's use of the cited approach is required. Exception requests will be approved by the appropriate authority and retained for a minimum of 5 years for use during milestone reviews/audits or pending waiver requests.

a. COTS software and vendor update implementations may be used with an exemption request. The COTS may neither be modified in function nor maintained by the government. (The policy regarding the use of COTS software packages (e.g., DBMSs, graphics) to generate application programs that are not in Ada is addressed in Advanced Software Technology.)

b. Software which has already been operationally fielded may be reused with an exception request subject to the following conditions: (1) The existing source code is written in a standard HOL; (2) The source code modified is less than 1/3 of compilable source code. (Modified code is the sum of code changes and additional codes. The 1/3 change will be assessed against the smallest unit of delivery (2167-CI, 7935-Subsystem Specification) and (3) use of assembly language is identified and limited to functions required to allow the standard HOL software to run on the targeted hardware.

c. Use of SQL (FIPS 127-1) with DBMSs for binding to Ada host applications is an Ada policy compliant approach with an exception request.

d. Use of non-Ada for special-purpose application processors (signal processors, array processors, FFT processors, etc.) provided that Ada is used for the command processor or general-purpose processor that directs the application is allowed subject to an exception request.

e. Non-Ada code may be used for a rapid prototyping project with an exception request. The project must be converted to Ada prior to operational implementation.

8. Waivers.

a. With the exceptions noted above, 85% or more of the compilable source code developed must be in Ada or else a waiver must be obtained.

b. Waivers are not required for development of new Ada code or reuse/modification of existing Ada code.

9. Procedures

a. Exceptions

(1) Milestone Decision Authority (MDA) is the approval authority for policy exceptions for programs under references (a) and (b). Chief of Navy Research is approval authority for policy exceptions for programs under reference (c).

(2) Exception requests shall be submitted to the MDA via the appropriate chain of command. The Ada Exception Notification format is provided in attachment (a).

(3) System acquisition and/or software development may proceed upon receipt of an endorsement from the MDA approving the exception.

b. Waivers

(1) Commander, Navy Information Systems Management Center (NISMC) is the approval authority for waivers to policy contained in this instruction.

(2) Waiver requests shall be submitted to Commander NISMC via the appropriate chain of command. The Ada waiver request format is provided in attachment (b).

(3) Waivers must be approved by Commander, NISMC before release of the final Request for Proposal for contractor software development and before system design begins for in-house development.

10. Responsibilities

a. ASN(RDA) shall:

(1) Establish Ada policy for the DON.

(2) Maintain oversight of the DON Ada Program to insert Ada-related technology into DON systems.

b. Deputy, Assistant Secretary of the Navy, Command, Control, Communications and Computers, Intelligence/Electronic Warfare/Space, DASN(C4I/EW/Space), shall:

(1) Review Acquisition Programs for compliance with this policy.

(2) Ensure that the policy and procedures in this instruction are implemented.

c. Commander, Navy Information Systems Management Center (NISMC) shall:

(1) As DON Ada Waiver Approval Authority, make final disposition on all Ada waiver requests.

(2) As the DON Software Executive Official in support of ASN(RDA), serve as the focal point for all Ada program activities and maintain the DON Ada Implementation Plan.

d. Chief of Naval Operations (CNO), Chief of Navy Research (CNR), Assistant for Administration for the Under Secretary of the Navy (AAUSN), Commandant of the Marine Corps (CMC) shall:

(1) Conduct one time review by 30 September 1992 to ensure compliance with this instruction within subordinate organizations. Submit the results of that review to Commander, Navy Information Systems Management Center by 30 October 1992.

(2) Ensure that all activities responsible for systems acquisition and/or software development have established Ada implementation guidance within 90 days of issuance of this instruction.

e. Milestone Decision Authorities shall:

(1) Make final disposition on all Ada exception requests.

(2) Retain Ada exception requests for a period of five years.

f. The Chief of Naval Research during the period of this interim policy shall:

(1) make final disposition on Ada waiver requests submitted from within his organization.

(2) at the end of the interim policy period, make a one time report to DASN (C4I/EW/S) advising him of the need to revise this policy to meet the needs of the laboratory community.

Ada EXCEPTION NOTIFICATION FORMAT

Cover Letter. An exception request must include a cover letter (not to exceed three pages), signed out by the proper releasing authority in the chain of command, to the Milestone Decision Authority. The cover letter should include at a minimum, the focal point (name, office symbol and phone), an identification of specific exemption being claimed, the details required by Exception Request Content described on next page, a statement identifying the responsible maintenance activity (in-house or contractor) associated with the software involved with the exception request, and a brief summary of the contents of the package. Additional details may be included in attachments to the cover letter.

Attachment (A)

EXCEPTION CONDITIONS/
EXCEPTION REQUEST CONTENT REQUIREMENTS

Condition 1. COTS software and vendor update implementations may be used with an exemption request. The exception request will list the commercial software being used for the system. The program office will certify that COTS is neither being modified in function nor maintained by the government.

Condition 2. Reuse and upgrade of existing DOD and government maintained software that meets the following criteria: (1) The source code is written in a HOL approved in DOD 3405.1; (2) The source code modified is less than one-third of the compilable source code (The one-third change will be assessed against the smallest unit of delivery.); and (3) The use of assembly language is identified and limited to functions required to allow the standard HOL software to run on the targeted hardware. An exception request must include the following information: Description of reused software, function, programming language(s), source lines of code, anticipated modifications, and software support activities aligned for current and modified software. Provide a description of Ada transition efforts and a statement of maintenance support.

Condition 3. An exception request is needed for non-Ada code written for special purpose processors (signal processors, array processors, etc.,) provided that Ada is used for the command processor or general-purpose processor that directs the application. Exception requests will identify the command and special purpose processors being used, the programming languages being used and their purpose, and the number of source lines of Ada code and special purpose code.

Condition 4. The exception request for use of SQL (ANSI, FIPS 127-1) with SQL compliant DBMSs will identify the commercial DBMS being used and the source lines of code for SQL and Ada being used for the application.

Condition 5. Rapid prototyping for the purposes of specifying determining or refining requirements, as long as the project is implemented in Ada. Evolutionary prototyping, for the purpose of incremental system development, must be done in Ada. An exception request must describe the rapid prototyping effort, non-Ada language used, and the Ada transition plan.

Attachment (2)

Ada WAIVER REQUEST FORMAT

Cover Letter: A waiver request package must include a cover letter (not to exceed one page), signed out by the proper releasing authority in the chain of command, to the approval authority Commander, NISMC. Cover Letter should include a focal point (office symbol and phone) and a brief summary of the contents of the package. The details are to be included in the attachments to the cover letter. The package must include the subparagraphs below and may not exceed ten pages in length.

Attachment 1, Executive Summary: This attachment includes a description of the capabilities needed, rationale and justification for not using Ada (to include cost, schedule performance, reuse, portability and risk), a description of the proposed system (hardware, software, firmware) and justification and rationale for selecting the proposed system.

Attachment 2, System/Project Descriptions: This attachment includes details of the proposed system, to include acquisition and contracting status (to the extent it is pertinent to the waiver decision), and description of both host and target hardware, software and firmware.

Attachment 3, Life Cycle Cost Analysis: This attachment provides a cost and benefit analysis which clearly shows that the proposed solution is more cost effective and beneficial to DON over the project's life than Ada. The analysis must address both the Ada solution and the proposed solution and include software development costs, life cycle maintenance costs, replacement costs, training, portability, reuse, productivity, performance, useability, documentation, interfaces, schedules, and higher authority program direction.

When computing the life cycle cost of an Ada solution, any initial investment in Ada support environments, tools, training, etc., must be amortorized over all future anticipated Ada projects. In such cases the amortized amount of the total investment should not exceed fifty percent, since the investment would be used for future projects.

Attachment 4, Transition Plan: This attachment describes your future plans for moving to Ada if the waiver is approved. Address all applicable factors, including language features, compilers, environments, bindings, training, education, schedules, personnel, costs and hardware.

Attachment (b)

Attachment 5, Risk Analysis: This attachment describes risks such as schedule, performance, security and other non-economic issues associated with both the Ada and non-Ada solutions.

Attachment 6, Statement of Maintenance: This attachment (limited to one page) must identify the responsible maintenance activity (in-house or contractor) associated with the software involved with the exception request.

Attachment (b)

APPENDIX C

OUTLINE FOR AIP AS OF SEPTEMBER 27, 1990

Ada Implementation Plan

Draft Outline with tentative personnel assignments

[Editor's note: this plan has a strong handbook flavor. Some though needs to be given to identifying its intended audience and the message they are to receive.]

EXECUTIVE SUMMARY

1.0 INTRODUCTION

[Clive Harding with everybody]

1.1 Purpose

1.2 Scope
- Applicable systems
- Acquisition phases

1.3 Assumptions

1.4 Requirements

1.5 Background

1.6 DON Ada Management Organizations
- DOD
- SECNAV
- Navy and Marine Corps

2.0 POLICY

[Cdr Romeo with Capt Despasquale]

- Ada advantages
- Policy rationale
- Policy description
- Waivers

3.0 PROGRAM MANAGER ADA IMPLEMENTATION GUIDANCE

[George Robertson with Robert Calland, Dan Green, Marshall Potter, and Toni Stuart]

3.1 Program Planning

- Cost and Schedule Estimation (development and life cycle)
- Resource requirements (development and life cycle)
- Role of program office
- Role of Navy laboratories
- Training
- How and when to obtain assistance

3.2 Acquisition Planning

- Technical requirements
- Work requirements
- Proposal content requirements
- Proposal evaluation criteria

3.3 Systems Engineering

- Role of Ada (development and life cycle)
- Risk management (planning, assessment, analysis, handling)
- Tradeoffs (money, time, capability, quality)
- Technical performance measures
- Effect of Ada on:
  - -- Reliability and Availability
  - -- Commonality
  - -- Hardware sizing and timing
  - -- Interfaces to existing systems
  - -- Prime/subcontractor relationships
- Scalability issues:
  - -- small-scale systems
  - -- medium-scale systems (>50K SLOC)
  - -- large-scale systems (>500K SLOC)

3.4 Software Engineering

- Role of Ada (development and life cycle)
- Software development metrics
- Development techniques (prototyping, inspections, etc.)
- Verification, Validation, and Acceptance
- Special concerns:
  - -- Ada PDL
  - -- Ada design and coding practices
  - -- CASE tools and Ada compilers
  - -- multiple languages and computer types
  - -- COTS (quality, legal, and life cycle)

- Categories of software:
  -- Operational (end-use)
  -- Simulation/Stimulation
  -- Program generation and support

3.5 Test and Evaluation
- Role of Ada (development and life cycle)
- (Ada's impact on schedule, quality, integration)

3.6 Integrated Logistics Support
- Role of Ada (development and life cycle)
- (Ada's impact on the ISEA and LCSA)

4.0 ADA ENVIRONMENTS

[Hank Stuebing with CDA, Frank Erwin, and Capt Thompson]

4.1 Mission Critical Computer Resources
- SECR ALS/N
- COTS Ads

4.2 Administrative Information Systems

4.3 Program Support Environments
- CASE
- Tools

5.0 CROSS-CUTTING ISSUES

[Shirley Peele with Rich Bergman, CDA, and Cdr Romeo]

5.1 Compilers
- Validation (ACVC)
- Evaluation (ACEC)
- Selection
- Vendor differences

5.2 Ada Secondary Standards
- Role of Ada bindings
- Operating systems
- Databases
- Graphics
- Windowing Environments
- Software Development Tools
  (library management tools, source level symbolic debuggers, program viewers, Ada-oriented editors, static and dynamic analyzers, CASE, source reformatters, cross referencers, and recompilation analyzers)

5.3 Ada transition
- Ada upgrade opportunities
- Reverse engineering

5.4 Life cycle documentation
- 2167A and HDBK-287
- 2167A tools
- 7935

6.0 LESSONS LEARNED

[Ron House with Rich Bergman and Capt Despasquale]

6.1 AFATDS

6.2 BSY-2

6.3 ALS/N

6.4 C2P Ada Shadow

6.5 CAC Reports

... about 3-4 others

7.0 FUTURE DIRECTIONS

[Tom Conrad with Cdr Romeo and Toni Stuart]

7.1 Next Generation Computer Resources

7.2 ALS/N

7.3 Ada 9X

7.4 DODD 5000.1

7.5 Software Master Plan

7.6 STARS

7.7 MIL-STD-1838 (CAIS)

APPENDIX A HELPFUL SOURCES (not in any order)

[Cathy Ruiz with Joan McGarity and CDA]

- Ada Joint Project Office
- Software Engineering Institute
- DON-IRM
- SPAWAR

- Software Productivity Consortium
- Institute for Defense Analysis
- PMS412
- Air Force Wright-Patterson ??
- Army CECOM ??
- AdaJUG
- Navy Ada Users Group
- Ada Information Clearing House

APPENDIX B USEFUL REFERENCES

[Software Master Plan Style]

- Policy
- Standards
- Guidance

APPENDIX C NAVY ADA PROJECTS [ALL NAVY TASK FORCE MEMBERS]

[AJPO style]

APPENDIX D MARINE CORPS ADA PROJECTS [ALL MARINE CORP TASK FORCE MEMBERS]

[ADPO style]

APPENDIX E GLOSSARY

[Clive Harding]

APPENDIX F USER UPDATE HOTLINE

APPENDIX TBD ADA RELATED ISSUES (not in any order)

[Robert Calland]

4.1 What to look for in your prime contractor

4.2 What to look for in your subcontractors

4.3 What to look for in your Navy laboratories

4.4 Understanding the Ada development cycle
- Tailoring/modifying 2167A
- Tailoring/modifying 2168

4.5 Why training is so critical

4.6 What areas require special attention?
- compiler vendors
- CASE tool vendors
- Software Development Plan

APPENDIX D

OUTLINE FOR AIP AS OF FEBRUARY 7, 1991

Department of Navy Ada Implementation Plan

This plan provides guidance to Program Managers and their staffs on implementing Department of Navy policies and standards for use of the Ada programming language. For the most part, guidance will be specific to Ada and assume some previous experience with software program management.

	Executive Summary	1 page, short paragraphs, wide margins		
1.0	Introduction	Formal	4 pages	PM
2.0	Policy	Formal	4 pages	PM & staff
3.0	Program Manager Ada Implementation Guidance	Handbook	15 pages	PM & staff
4.0	Ada Environments	Handbook	10 pages	PM Engineers
5.0	Ada Technology Issues	Handbook	15 pages	PM Engineers
6.0	Lessons Learned	Narrative	20 pages	PM Staff
7.0	Future Directions	Narrative	10 pages	PM Staff
A	Helpful Sources	(Organizations, Newsletters, Bulletin Boards)		
B	Useful References	(patterned after DoD S/W Master Plan Part 2)		
C	Glossary			

D	Navy Ada Projects			
E	Marine Corps Ada Projects			
F	Dept of Navy Training Plan			
G	PPBS			

APPENDIX E

OUTLINE FOR DON ADA TRAINING PLAN AS OF FEBRUARY 7, 1991

DON ADA TRAINING PLAN

1. INTRODUCTION

   - Discuss rationale for Ada training
   - Discuss importance of developing organic resources

2. REQUIREMENT

   - Explain PL 8084
   - Meet software development functional requirements, schedules, and budgets
   - Reduce Post Deployment Software Support costs

3. TRAINING APPROACHES

   - Formal (This section will address the course material and the target audience)

     -- Top Management Overview (Executive Seminar)
     -- Program Manager Introduction
     -- Project Management/Cost Estimating
     -- Software Engineering
     -- Object Oriented Program Design
     -- Fundamentals of Ada Programming
     -- Advanced Ada Programming Concepts and Techniques
     -- Ada Development Support Environment (CASE Tools)

   - Informal

     -- Mentors
     -- CBT/CAI
     -- Programming Teams (Projects)

4. TRAINING SOURCES

   - Academia
   - Consultants
   - Service Schools (NEC)
   - Other DoD Courses
   - In-house Training Programs

5. TRAINING PLAN DEVELOPMENT AND EVALUATION

- Length/Topics
- Environment/Equipment
- Hands-on Lab
- IS Projects
- Availability of Mentor/Instructor
- Track Actual vs. Planned IS Functionality, Schedule, and Budget
- Document Feedback from Staff

6. LESSONS LEARNED

- MCCR Community
- MIS Community
- Scientific Community
- Private Sector
- Ada Joint Program Office
- Software Engineering Institute

7. FUNDING CONCERNS/SOURCES

APPENDIX F

TASK FORCE MEMBERS AS OF JUNE 20, 1991

DEPARTMENT OF NAVY
ADA IMPLEMENTATION GUIDE
TASK FORCE PARTICIPANTS

CHAIR: DASN(IRM)	Ms. Antoinette Stuart
DEPCHAIR: SPAWAR	CDR Martin Romeo
AJPO	Mr. Currie Colket
FCDSSA, San Diego	Mr. George Robertson
FCDSSA, Dam Neck	Ms. Shirley Peele Mr. Guy Taylor
FMSO	Mr. Lester Hummel
NADC	Mr. Hank Stuebing Mr. Chuck Koch
NADEP	Mr. John McLaurin
NAVAIR	Mr. Tom Coyle
NAVCOMTSSA	Mr. Jim Welch
NAVSEA	Mr. Greg Engledove
NCTC	Ms. Joan McGarity
NOSC	Mr. Robert Calland Ms. Cathy Ruiz Mr. Rich Bergman Ms. Donna K. Fisher
NSWC	Mr. Dan Green Mr. Frank Ervin Mr. Eugene Hodgson Mr. Charles Flemming

NUSC	Mr. Ron House Mr. Tom Conrad
USMC	Capt Gerald DePasquale Capt Dave Thompson
USNA	Mr. Doug Afdahl Mr. Jim Moss Major J. Spegele
NARDAC, San Francisco	Ms. Patricia Grandy
NAVCOMTELSTA	Mr. Bond Wetherbe Mr. George Frilot
Navy Postgraduate School	LCDR Jean Shkapsky
NATC	Ms. Kathy Steele Mr. John Shields
BUPERS	LCDR Anne Sullivan
Booz, Allen & Hamilton	Ms. Susan Scott

LIST OF REFERENCES

Anthes, Gary H., "Ada Making its Mark at Commercial Sites," Computer World, June 17, 1991, Vol. XXV, No. 24.

Boehm, B.W., "Software and Its Impact: A Quantitative Assessment," Datamation, May 1973.

Booch, Grady, Software Engineering with Ada, Benjamin/Cummings Publishing Company, Inc., 1986.

Department of the Air Force Action Memo of January 29, 1989, Assistant Secretary of the Air Force (Acquisition), Subject: AF Ada Implementation Plan.

Department of the Navy Ada Implementation Guide, AIP Task Force, draft, 1991.

Department of the Navy Ada Implementation Plan Task Force Minutes, November 1991.

Education and Training Plan: Department of the Navy Software Engineering Using Ada, draft, 1991.

Emery and McCaffrey, "Policy Issues Concerning Implementing the Ada Mandate within the Department of Defense," draft, May 1991.

Engle, Charles B., "The Myth of Portability in Ada," Software Engineering Institute, Carnegie Mellon University, 1991.

Fisher, D.A., "A Common Programming Language for the DOD-Background and Technical Requirements," Report P-1191, June 1976.

IEEE Computer Society Board of Governors, "Society's Board Backs Report Urging Upgrading Computer Science Schooling," IEEE Computer, September 1988.

IIT Research Institute, "Estimating the Cost of Ada Software Development," IIT Research Institute, Lanham, MD, April 1989.

Kitfield, James, "Is Software DOD's Achilles' Heel?", Military Forum, July 1989.

Knight, K.L., "Assessing the Impact of Implementation of the Ada Programming Language at Naval Regional Data Automation Centers," December 1990.

Lewin, Kurt, "Frontiers in Group Dynamics: Concept, Method and Reality in Social Science," Human Relations 1, No. 1, 1947.

Mensching and Adams, Managing an Information System, Prentice Hall, 1991.

Reifer, D., "Economics of Ada---Guest Column," Ada Strategies, Vol. 5, No. 1, January 1991.

Skansholm, Jan, Ada From the Beginning, Addison-Wesley Publishing Company, 1988.

Spegele, J.J. and Park, E.K., "Adopting Ada as a Primary Undergraduate Programming Language," 9th Annual National Conference on Ada Technology, 1991.

Subcommittee on Investigations and Oversight, "Bugs in the Program: Problems in Federal Government Software Development and Regulation," September 1989.

U.S. Congress, Department of Defense Appropriations Act, 1991, Public Law 101-511 (November 5, 1990), 104 Stat. 1856-1914.

U.S. General Services Administration Information Resources Management Service, Overview Guide, January 1990.

Whitaker, W.A., "The U.S. DOD Common HOL Effort," SIGPLAN Notices, 1978.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey, California 93943-5002	2
3.	LCDR Roger Stemp, Code OR/St Department of Operations Research Naval Postgraduate School Monterey, California 93943-5000	1
4.	Professor Tung X. Bui, Code AS/Bd Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
5.	CDR Tom Hoskins, Code 37 Curricular Office for Computer Technology Naval Postgraduate School Monterey, California 93943-5000	1
6.	LCDR Jean Marie Shkapsky 1081 Wranglers Trail Pebble Beach, California 93953	1

Thesis
S4776 Shkapsky
c.1 Analysis of training-related issues in the transition to Ada in the DON.